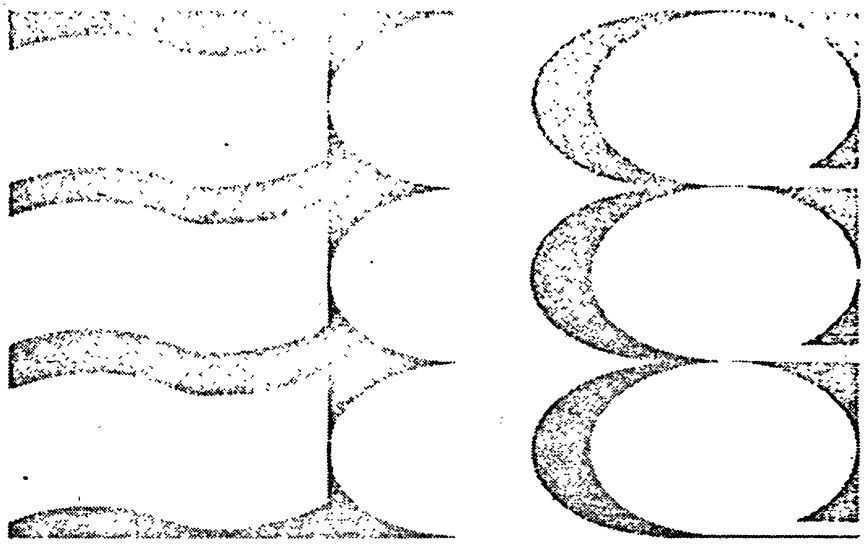

UNIVAC ENGINEERING PROGRAMMING LIBRARY

ENGINEERING PROGRAMMING SECTION - DATA PROCESSING DIVISION, ROSEVILLE, MINNESOTA

9400 MASTER CHANNEL ADAPTER (F1001-01) TEST
PROGRAM DESCRIPTION DRAWING

4091649

DRAWING NUMBER 4091649

REVISION A

UNIVAC

ENGINEERING PROGRAMMING LIBRARY

9400 MASTER CHANNEL ADAPTER (F1001-01) TEST PROGRAM DESCRIPTION DRAWING

	SIGNATURE	TITLE	DATE
APPROVAL			
APPROVAL			
APPROVAL			
APPROVAL			

Associated Listing Revision is: A

REVISION STATISTICS RECORD 4091649

DWG REV	MSTR REV	RELEASE DATE	DESIGNER NAME	EIR NUMBER	FCO NUMBER
-	F	10/10/69	W. R. Karban	A89798	30190029
A	G	3/06/70	W. R. Karban	A89800	30190030

REVISION DESCRIPTION RECORD

REV	DESCRIPTION
-	Release

REVISION DESCRIPTION RECORD 4091649

REV	DESCRIPTION
-	Release
A	Retype documentation to correct typing errors, table of contents etc. Added checking to tests 21 through 26 to check for 'reject' in the sense byte if status contains unit check.

4091649

Table of Contents

4091649

9400 MASTER CHANNEL ADAPTER (F1001-01) TEST PROGRAM DESCRIPTION

1.0 INTRODUCTION

1.1 Purpose - This document describes the 9400 Master portion of the test which check the Channel Adapter 9000 (F1001-01).

1.2 Major Objectives - This MAR is designed to perform a functional test of the 9000 Series Channel Adapter when used in a 9400-9200/9300 configuration.

1.3 Equipment Configurations - This MAR is used with the following equipment configuration:

9400 Processor with Multiplexer (MPX) I/O Channel
9000 Series Channel Adapter
9200/9300 Processor with General Purpose I/O Channel

It is important to note that the 9400 Processor is not capable of operating the Channel Adapter on a Selector Channel.

1.4 Associated Software - This program relies on the Maintenance Control Routine (MCR) to perform program load, parameterization, deletion, and I/O initiation and handling when not in the Timing Mode. It is used with the 9300 Channel Adapter Test Program (Drawing Number 4091649) to complete the interface checkout of the Channel Adapter 9000.

1.5 Reference Documents - Documents used as reference material for the developments of this test program are listed below.

Drawing	Revision	Description
SU0039		9000 Series I/O Interface Equipment Specification
S-90065		F1001-01 Channel Adapter 9000 Product Description
S-70040		9400 Processor and Console Product Description
4091622		9400 Maintenance Control Routine Program Description Drawing
4091649		9300 Channel Adapter Test Program Description
SA00823		9400 System Equipment Specification
		Technical Memorandum 699-01
		Technical Memorandum 699-02

1.6 Notes and Restrictions - Although the Channel Adapter 9000 is capable of operating on either a Multiplexer or a Selector Channel, it is accepted only on the MPX Channel when operating in a 9400 configuration.

Program Option V1 (Print Error Messages) is available on load and may be deleted at the operator's descretion.

The Standard Program Option V3 (allow Error Recovery three times) is built into this program and is not considered as an independent option in this MAR.

If Program Option V1 has been deleted and the test encounters an error, it will proceed as though no error has occurred. There will be no error message and no error recovery attempt.

The Resume parameter may be used if the program has been stopped via the End parameter (and no errors). Any other use of the Resume parameter may have unpredictable results.

When the 'Master' program starts a test sequence with the Slave program, the first command issued should result in an 'Attention' interrupt to the Slave. The Slave program responds with a Sense Command to obtain the sense byte, which contains the command and detail bits. The Slave program selects a test by interpreting the sense byte. If the sense byte is incorrect, the two programs may become 'hung'. The adapter cannot be arbitrarily cleared by using a test I/O override command so it is necessary to restart the Master program with a Begin parameter, and restart or reload the Slave program.

The MCR does a 'sense' if status contains 'unit check' and the problem program is not in timing mode. This sense may not appear in the test description as it is not performed by the problem program.

2.0 FUNCTIONAL DESCRIPTION

2.1 General Organization - This MAR interfaces a similar program in the 9200/9300 Processor. Each test program controls its respective processor in checking out the Channel Adapter 9000 (CA 9000) when used in a 9400 - 9200/9300 configuration. The tests are constructed to coordinate their activities, ie.e., when this MAR transmits (either a command or data) the complementing test program will send the appropriate response if the CA 9000 is working properly.

Testing sequence is in numerical order from lowest (T2) to highest (26). Each test has a specified location in the test list and, consequently, although the tests may be entered out of sequence they will still be run in numerical order. Specific parameter entry techniques are described in paragraphs 3.4.1 and 3.4.2.

2.2 Test Descriptions - Each of the following tests is described by stating its objective and the method used to attain it.

2.2.1 Test 1 (Undefined) - Test 1 remains undefined at this time.

2.2.2 Test 2 - Test No-Op command in 'Idle' mode for the 9400 Master program.

Objective: Test that the adapter accepts the No-Op command and returns Channel End - Device End Status.

Method:

1. A No-Op command is sent to the adapter.
2. Status is checked for Channel End - Device End.
3. If the No-Op command is not accepted or the status is incorrect, the above procedure is repeated 3 times before exiting from this test.

2.2.3 Test 3 - Test Sense command in 'Idle' mode for the 9400 Master program.

Objective: Test that the adapter accepts the Sense command 'Idle' mode, returns status of Channel End - Device End and the sense byte = 00.

Method:

1. A Sense command is sent to the adapter.
2. Status is checked for Channel End - Device End.
3. The sense byte is checked for a value of ØØ.
4. If the Sense command is rejected or steps 2 or 3 detect an error, the above procedure is repeated 3 times before exiting from this test.

2.2.4 Test 4 - Test the 'Illegal' command in 'Idle' mode for the 9400 Master program.

Objective: Test that the adapter rejects the Illegal command, returns a status of unit check and the sense byte contains Command Reject ($8\emptyset_{16}$).

Mode:

1. The Illegal command ($\emptyset 8_{16}$) is sent to the adapter.
2. Status is checked for the unit check bit set ($\emptyset 2_{16}$).
3. If the status was 'unit check' the program executes steps 1 and 2 three times before exiting from the test.
4. The sense byte is checked for the Command Reject bit set (80_{16}).
5. Any error causes the test to go to step one.

The above steps are repeated a maximum of three times before exiting from this test.

2.2.5 Test 5 - Minimum test for Read Command. ('Slave' Test 2)

Objective: Test that the adapter accepts the Read Command, the Read Command causes attention status to the Slave program and a Write Command from the Slave causes a 'Match' resulting in a 1 byte transfer.

1. The Master program issues a Read Command.
2. If the Slave program responds properly, a 'Match' condition should occur.
3. Master status is checked for Channel End - Device End.
4. Transfer of 1 byte of data is verified by checking the BCW byte count in the 9400 Master.
5. There is no error recovery.
6. The data is not verified.

2.2.6 Test 6 - Minimum test for Read Backward Command. ('Slave' Test 2)

Same as for test 5 except a Read Backward Command is used by the 9400 Master program.

2.2.7 Test 7 - Minimum test for Write Command. ('Slave' Test 3)

Same as for test 5 except a Write Command is used by the 9400 Master program. The output byte - 03_{16}.

2.2.8 Test 8 - Test Control Command. ('Slave' Test 4)

Objective: Test that the adapter accepts the Control Command with all combinations of detail bits and returns status of Channel End - Device End.

Method: An index is set to FC_{16} and is used to form the detail bits. The Control Command is 'ored' with the index to form the command sent to the adapter. For each subsequent operation, the index is decremented by 1 and bits 6 and 7 are 'Anded' off, since the Control Command occupies those bits. The last command sent = 07_{16}.

1. A Control Command and detail bits are sent to the adapter.
2. Status is checked for Channel End - Device End. Correct status is interpreted as meaning the Control Command was completed successfully.
3. Steps 1 and 2 are repeated until the index is decremented to zero.
4. A status error will cause an exit from this test.

2.2.9 Test 9 - Test Test I/O and Test I/O Override Commands in 'Idle' mode for the 94ØØ Master program.

Objective: Test that the adapter accepts the Test I/O and Test I/O Override Commands in 'Idle' mode and the condition code = Ø after executing each command.

Method: The test requests timing mode from the MCR then proceeds as follows:

1. A one word buffer is set up.
2. A Test I/O command is sent to the adapter.
3. The condition code is checked for Ø.
4. The BCW is checked for no change in byte count.
5. A Test I/O Override Command is sent to the adapter.
6. The condition code is checked for Ø.
7. The BCW is checked for no change in the byte count.
8. If any error occurs, the program exits from this test.

2.2.10 Test 10 - Data Test #1. ('Slave' Test 5)

Objective: Test transfer of alternate bytes of 'ones' and zeroes (FF00FF00...).

Method:

1. A Read Command is sent to the adapter. The BCW byte count is 16Ø bytes.
2. Ending status is checked for Channel End - Device End.
3. If any data was transferred, those bytes are verified for the correct value.

4. The program exits from this test after completing steps 1-3.
5. There is no error recovery.

2.2.11 Test 11 - Data Test #2 ('Slave' Test 6)

Objective: Test transfer of bit-no, bit data (AAAA...).

Method: Same as for test 10 paragraph 2.2.10 except for the data pattern.

2.2.12 Test 12 - Data Test #3 ('Slave' Test 7)

Objective: Test transfer of all 256 bit combinations possible in 1 byte (compress mode characters).

Method: Same as for test 10 paragraph 2.2.10 except four data buffers of 80 bytes each are used. The last buffer is padded with zeroes.

2.2.13 Test 13 - Data Test #4 ('Slave' Test 8)

Objective: Transfer data specified by the operator.

Method: Same as for test 1Ø paragraph 2.2.10 except the data pattern is derived fror 2 bytes of data entered via V6. The 2 bytes are repeated through the 16Ø byte buffer, i.e., if the data entered = 3F4C the data buffer would = 3F4C3F4C...3F4C.

2.2.14 Test 14 - Data Test #5 ('Slave' Test 9)

Objectives: Test transfer of all 'ones' (FFF---).

Method: Same as for test 1Ø paragraph 2.2.10 except for the data pattern.

2.2.15 Test 15 - Data Test #6 ('Slave' Test 10)

Objective: Test transfer of all 'zeroes'.

Method: Same as for test 1Ø paragraph 2.2.10 except for the data pattern.

2.2.16 Test 16 - Test I/O Override test #1. ('Slave' Test 11)

Objective: Test that the Test IO Override Command clears the adapter after a Read Command has been issued.

Method:

1. A Read Command is issued.
2. 'Slave' program responds with a Test I/O Override Command.
3. The program checks for ending status of Channel End - Device End and Unit check. The sense byte is checked for reject ($Ø2_{16}$).

2.2.16 (cont.)

4. The program exits from this test after completing steps 1-3. There is no error recovery.

2.2.17 Test 17 - Test I/O Override test #2 ('Slave' Test 12)

Objective: Test that the Test I/O override command clears the adapter after a Read Backward command is issued.

Method:
1. A Read Backward command is issued.
2. 'Slave' program responds with a Read Backward command then a Test I/O Override command.
3. The program checks for ending status of Channel End-Device End and Unit Check. The sense byte is checked for Reject (Ø2).
4. The program exits from this test after completing steps 1-3 there is no error recovery.

2.2.18 Test 18 - Test I/O Override test #3 ('Slave' Test 13)

Objective: Test that the Test I/O Override command clears the adapter after a Write command is issued.

Method: Same as for test 16 paragraph 2.2.16 except a Write command is issued instead of a Read command.

2.2.19 Test 19 - Test 'Test I/O Override' in Data mode. ('Slave' Test 14)

Objective: Test that the adapter is cleared by a Test I/O Override if the adapter is in 'Data' mode.

Method: Timing mode is requested from the MCR. When the request is granted the program proceeds as follows:

1. A 'Read' command is issued with a buffer length of 16Ø bytes. The condition code is checked for zero.
2. The program checks for a data transfer by monitoring the BCW byte count.
3. When the byte count changes it is assumed a data transfer is in progress and a Test I/O Override command is issued. The condition code is checked for 1.
4. Ending status is checked for Channel End - Device End.
5. Any error causes an exit from this test. There is no error recovery.

2.2.20 Test 20 - 'Immediate Busy' Test.

Objective: Test that the adapter returns an 'Immediate Busy' status ($5\emptyset_{10}$) if the previous command is not completed.

Method: The program sets the CCB so a flag is set in the CCB if an interrupt occurs. This is so the test can determine when the Slave program has sent a Test I/O Override command.

1. A No-Op command is sent to set the CCB as explained above.
2. Timing mode is requested from the MCR.
3. A Write command is issued with a data buffer of $16\emptyset$ bytes. The condition code is checked for zero.
4. A Read command is issued and status is checked for 'Immediate Busy' ($5\emptyset_{16}$).
5. The program enters a delay loop to check for the interrupt caused by the Test I/O Override command from the adapter. One second is allowed. The CCB is then set so the MCR will consider any unexpected interrupt as NSI's and the MCR will print its NSI message if such an interrupt occurs.
6. The program exits after any error. If such an exit occurs before the Slave program sends its Test I/O Override, an NSI message will be initiated by the MCR.

2.2.21 Test 21 - Slave program 'Idle' mode test #1. ('Slave' Test 16)

Objective: Allow the Slave program to test the No-Op command in 'Idle' mode.

Method:

1. A Read command is sent to the adapter.
2. The Slave program should send a Test I/O Override command to put the adapter in 'Idle' mode.
3. Ending status is checked for Channel End - Device End - Unit Check.
4. The Sense byte is checked for 'reject' ($\emptyset 2$).
5. A one second delay is initiated to allow the Slave program time to check the No-Op command.

2.2.22 Test 22 - Slave program idle mode test #2. ('Slave' Test 17)

Objective: Allow the Slave program to test the Sense command in 'Idle' mode.

Method: Same as for test 21 paragraph 2.2.21 except the Slave program tests the sense command.

2.2.23 Test 23 - Slave program Idle mode test #3. ('Slave' Test 18)

Objective: Allow the Slave program to test the Illegal command in 'Idle' mode.

Method: Same as for test 21 paragraph 2.2.21 except the Slave program tests the Illegal command.

2.2.24 Test 24 - Control command test for the Slave program. ('Slave' Test 19)

Objective: Test the Control command issued by the Slave program.

Method:

1. A Read command is issued and a flag is set in the CCB so the MCR does not print an NSI message for this program.
2. The Slave program sends a Test I/O Override command to put the adapter in idle mode.
3. Ending status is checked for Channel End - Device End and Unit Check.
4. If Status contains Unit Check, the sense byte is checked for 'Reject' (02_{16}).
5. The program enters a one second delay loop and checks for the interrupt caused by the Slave programs issuing the control command.
6. If the delay expires before the interrupt is received, the CCB is set so the MCR will print an NSI message if the interrupt occurs and the program exits from this test.
7. If the interrupt is received before the delay expires, the CCB is set so the MCR will not print an NSI message. The program sends a Sense command to the adapter.
8. End status is checked for Channel End - Device End.
9. The sense byte is checked for a value of $\emptyset 7_{16}$.
10. There is no error recovery.

2.2.25 Test 25 - Slave program 'Idle' mode test #4. ('Slave' Test 15)

Objective: Allow the Slave program to test the Test I/O Override and Test I/O commands in 'Idle' mode.

Method: Same as for test 21 paragraph 2.2.21 except the Slave program tests the Test I/O and Test I/O Override commands.

2.2.26 Test 26 - Slave program 'Idle' mode test #5. ('Slave' Test 20)

Objective: Allow the Slave program to check for 'Immediate Busy' status ($5\emptyset_{16}$).

Method: Same as for test 21 paragraph 2.2.21 except the Slave program executes its 'Immediate Busy' test. The Master terminates the Slave 'Control' mode with a 'Test I/O Override'. Master status is checked for zero.

3.0 OPERATING PROCEDURES

3.1 Initialization - Prior to testing the operator must:

1 - Energize the equipment
2 - Load the Maintenance Control Routine (MCR).
3 - Select the operating parameters that he will use.

3.2 Program Loading - The procedure for loading this MAR is shown below.

1 - Press the ATTENTION key. The console responds with the @ sign, the time stamp, and a space and then waits for an operator entry.
2 - Type in "RU". The console responds by printing an "N" and a space to the "RU".
3 - Type in the program name S3030M.
4 - Press the EOM key (symbols). The load and run statement which initiates loading and running the program appears on the console as shown below.

@00:10 RUN S3Ø3ØM s

5 - When the program has been loaded, the MCR prints a message which indicates the program's job number and the starting address of t;he test program as shown below.

JOB 1, S3Ø3ØM LOADED AT 02990

3.3 MAR Parameters - The parameters, of Program Designators, used in this MAR are grouped in three classes according to function: Action Designator, Equipment Designators, and Test Designator. Each designator is designed and described in the subparagraphs below.

3.3.1 Action Designators - These designators inform the Parameter Analysis Routine (PAR) how the parameters which follow should be processed. The following Action Designators are recognized by this test program:

A = Add Designator - directs the program to add the following parameters
D = Delete Designator - directs the program to delete the following parameters
B = Begin Designator - directs the program to begin testing
E = End Designator - directs the to stop testing
R = Resume Designator - directs the program to resume testing
V = View Designator - Display parameters for the specified subchannel(s)

3.3.2 Equipment Designators - These designators define the subsystem being tested. The equipment designatorsrecognized by this MAR are defined as follows:
Sn = Subchannel number

3.3.3 Test Designators - These designators allow the operator to modify the MAR's test environment by selecting various test (Tn) and/or program options (Vn).

Tn specifies a test (range 1 thru 26) where
Vn specifies a program option where:
V1 = Print Error Messages (Available on load)
V2/n = Delay n milliseconds between operations
V3 - Attempt Error Recovery Three Times (Available on load)
V6 = Operator Specified Data (2 bytes)
V4 = Stop on Error after Printout
V7 = Direct Error Message to High Speed Printer

3.4 Program Modifying - This MAR is adjusted to a specific operating environment by entering unsolicited parameters. The following subparagraphs define and/or describe the general parameter statement format and illustrate specific procedures for entry and deletion of parameters.

3.4.1 Parameter Entries - Parameter statements are entered in the general format shown below.

d Sn Tn Vn/y # Nxx Ⓢ

Where:

Symbol	Description
D =	Action Designator
Sn =	Subchannel number (an Equipment Designator).
Tn =	Test (s) to be executed (Test Designator).
Vn/y =	Program Option Vn with an extension value of y.
/ =	Phrase extension separator.
space =	Phrase separator.
Ⓢ =	End-of-message symbol.

When entering parameters, the following procedure is used:

1 - Press the ATTENTION key. The console responds with an @ sign and a time stamp (@03:47) and waits for a run identifier entry.
2 - Type in the one digit job or run number and a comma. The job number entry must be made within two minutes after the ATTENTION key was pressed or an abort typeout will occur. An example of an entry statement to this point is shown below.

@ 03:47 1,

3 - Enter a maximum of 63 characters of parameter data, ending the statement with a carriage return. (The MCR prints a stop code when it detects a carriage return entry.) The operator must enter all of the data following the comma within 2 minutes or an abort message will be printed. A completed parameter statement is shown below.

@ 03:47 1, A S2 T2-5, 8 (S)

4 - Enter as many parameter statements as are necessary; then begin the program with a Begin Action Designator as shown below.

@ 03:50, B (S)

3.4.2 Parameter Notes and Restrictions - A parameter statement beginning with an Add Designator signals the MAR to add the following information to the appropriate parameter table. For instance, the parameter statement below adds Tests 6, 7, and 8 to the Subchannel 1 parameter table.

A S1 T6-8 (S)

The following parameter statement adds Test 2 and 5, Variable 2 with an extension value of 5, and Variable 4 to the Subchannel 5 parameter table:

A S5 T2, 5 V2/5,4 (S)

It should be noted that each test that is added to the initial test selection will be entered in its predetermined position (by numerical order) in the test list rather than at the end of the list. For instance, assume that the initial parameter statement is as follows:

A S2 T2, 6, 7, 8, 12. B (S)

Testing would begin with Test 2, proceed with Tests 6, 7, 8, and 12 in that order and return to Test 2. Any additional test entries will be placed in the test list in numerical order. For instance, if Tests 3, 5, 9 and 11 are added, the program will cycle through the tests in the following order: 2, 3, 5, 6, 7, 8, 9, 11, and 12.

3.4.3 Deleting Parameters - A parameter statement beginning with a Delete Designator causes the deletion of the specified test designators from the parameter table. If no test designators are specified, all test designators are deleted from the test table except those designators present in the original load condition of the program. For example: The following statement deletes Test 3 and Variable 2 for Subchannel 2:

D S2 T3 V2 (S)

If a deletion entry contains a test designator (Th or Vn), but does not contain a Subchannel number (Sn), it is illegal and is ignored. The following entries are illegal because they lack a Subchannel number:

D T3 (S) D V2 (S)

3.5 Program Starting - The program is started by the MCR as soon as it is loaded. When the program is started, it types out the message:

*A 00:15 1S3030M ENTER PARAMETERS

and waits for a parameter entry. Parameters are entered as described in paragraph 3.4.1.

3.6 Program Stopping - Entering a parameter statement beginning with an End Designator will stop the program in progress. It should be noted that End Designators apply only to subchannels and not to tests or variables. Three examples of the use of the End Designator are shown below:

The following statement stops the entire program:

E ⟨S⟩

The following statement stops a specific subsystem on a given subchannel:

E S3 ⟨S⟩

The following statement stops multiple subsystem on a given subchannel:

E S2, 5-6 ⟨S⟩

If all testing is discontinued either because all tests are deleted from a subchannel or because a test stopped on an error (V4 entered), the program ends that subchannel as though it had been stopped by an End parameter statement.

3.7 Program Restarting - The test program can be restarted by one of the following entries:

A parameter entry having the format:

B ⟨S⟩

This entry will restart the test program from its initial starting point.

A parameter entry having the following format:

R ⟨S⟩

This entry will restart the test program at the point at which it was stopped. It is effective only if the testing has been ended on that subchannel via the End parameter and there are no errors.

The begin and Resume Designators only apply to subchannels and do not apply to tests or variables.

3.8 Program Termination - The program may be removed from storage by submitting a Cancel directive to the MCR as shown below.

1 - Press the ATTENTION key. The MCR will type the time and wait for additional entries.

2 - Type the letters CA. The MCR will respond with the letters NCEL and a space. Typeout to this point appears as follows:

@ 15:25 CANCEL

3 - Type the job identifier. The MCR will respond with a stop code indicating the completion of the program termination. An example of a complete Cancel directive is shown below.

@ 15:25 CANCEL 1 ↵

3.9 MAR Messages - This MAR generates three types of messages: Parameter Error Messages, Subsystem Error Messages, and Information Messages. These messages are all displayed on the console unless Program Option V7 has been entered. The following subparagraphs define and describe the mnemonics and symbols used in the MAR messages, and provide an example and explanation of each message.

3.9.1 Mnemonics and Symbols - The mnemonics and symbols used in the MAR messages are defined below.

MNEMONICS		
S3Ø3OM	=	Program Name
Sn	=	Subchannel Number (1-7)
Tn	=	Test Number
PS	=	Previous Status
PC	=	Previous Command
CS	=	Current Status
CC	=	Current Command
ES	=	Expected Status
SB	=	Sense Byte
D	=	Declarative Message
I	=	Imperative Message
Q	=	Question Message
BT	=	Bytes Transferred
TB	=	Total Bytes checked (data)
TBB	=	Total Bad Bytes
ESB	=	Expected Sense Byte
CSB	=	Current Sense Byte
GB	=	Good Byte (data)
BB	=	Bad Byte (data)

SYMBOLS		
hh	=	hour (00-23)
mm	=	minutes (00-59)
r	=	run number
n or nn	=	numerical value of the mnmonic prefix

3.9.2 Parameter Error Messages - These messages result when incorrect or insufficient parameter entries are detected. This MAR prints the following Parameter Error Messages:

1. Lacks Subchannel

   Message: hh:mm r S3Ø3OM LACKS SUBCHANNEL

   Cause: There are no subchannels in the subchannel table.

   Program Action: The MAR prints the message and waits for more parameters.

   Operator Action: Enter parameters, specifying a subchannel.

2. Lacks Test

   Message: S3Ø3ØM SUBCHANNEL N LACKS TEST

   Cause: There is a subchannel in the subchannel table, but no test(s) for it.

   Program Action: The MAR prints the message and waits for more parameters on that subchannel.

   Operator Action: Enter parameters, assigning a test to the specified subchannel.

3.9.3 Subsystem Error Messages - These messages contain information about errors in the Adapter. When an error is detected, the MAR prints the error message if V1 (available on load) is active, attempts error recovery and then proceeds with the test. If V4 (Stop on Error) is active, the program stops and must be restarted with a Begin or Resume statement. The operator's action is variable and depends on whether or not he has employed V4, and if so, whether or not he wishes to continue with the test. The Subsystem Error Messages Generated by this MAR are shown below.

1. Stopped on Error

   Message: S3Ø3ØM Sn Tn STOPPED ON ERROR

   Cause: An error condition has halted the program for one of three reasons:

   1. V4 is selected for the indicated subchannel
   2. A Timeout has occurred
   3. A Start I/O Command is issued and no response is received from the Adapter.

2. Timeout

Message: S3Ø3ØM Sn Tn TIMEOUT

Cause: The MCR timed out the last I/O operation. The timeout time varies from 1.000 to 1.999 seconds.

3. Incomplete Data Transfer

Message: S3Ø3ØM Sn Tn INCOMPLETE DATA TRANSFER
BTnn of nn

Cause: This message indicates an incomplete data transfer and provides the total number of bytes transferred during the last I/O operation.

4. Data Error

Message: S3Ø3ØM Sn Tn DATA ERROR

GB = nn
BB = nn
TBB = nn TB = nn PC = nn CC = nn

Cause: The data transferred doe. not equal the expected data.

5. Status Error

Message: S3Ø3ØM Sn Tn STATUS ERROR

PS = nn
CS = nn
ES = nn
PC = nn
CC = nn
SB = nn

Cause: The status generated by the previous I/O operation is not the expected status.

6. Sense Byte Error

Message: S3Ø3ØM Sn Tn SENSE BYTE ERROR

ESB = nn
CSB = nn
CC = nn
PC = nn

Cause: The sense data received does not equal the expected sense data.

7. MCR Sense I/O Failed

Message: S3Ø3ØM Sn Tn MCR SENSE I/O FAILED
Sense Status = nn

Cause: An MCR initiated Sense I/O operation failed because a status byte with the Unit Check bit set was received.

8. Buffer Control Word Error

Message: S3030M Sn Tn Buffer Control Word Error
EXP BCW = nnnnnnn Act BCW = nnnnnnnn
PC = nn CC = nn

9. Condition Code Error

Message: S3030M Sn Tn condition code error
EXP Cond. Code = nn Act Cond. Code = nn
PC = nn CC = nn

Cause: Condition code was not as expected.

10. Timing Mode Terminated

Message: S3030M Sn Tn Timing mode terminated
MCR Release Code = nn
Channel and Device = nn

Cause: The MCR terminated timing mode as indicated by the MCR release code.

3.9.4 Information Messages - This MAR prints only one information message as shown below.

Message: S3Ø3ØM ADAPTER TEST INACTIVE

Cause: Either all 'RUN' flags have been cleared or no Begin parameter statement has followed a parameter entry with all subchannels inactive.

Program Action: The MAR prints the message and waits for more parameters.

Operator Action: Enter a Begin or Resume parameter statement as appropriate.

4.0 SUPPLEMENTARY DATA

4.1 Supplementary Hardware Information - The Channel Adapter 9000 (CA 9000) is an asychronous logic device that provides a half-duplex connection between two processor channels. It is capable of controlling data transfer over either a Multiplexer or a Selector Channel, but is restricted to Multiplexer operation by the Processor.

4.2 Supplementary Software Information - This portion of Section four deals with the operation of the Channel Adapter from a software point of view. In the following subparagraphs it describes 1) the three modes of operation and 2) the operation of the Adapter under error conditions.

4.2.1 Idle Mode - The Channel Adapter is in the Idle Mode when it has not recognized an Initial Selection Sequence from either channel or when it does not have Pending Status. Detailed information on the status reply to commands issued by either channel when in the Idle Mode is shown in Table 1 of the F1001-01 Channel Adapter 9000 Product Description.

4.2.2 Control Mode - The Channel Adapter is in the Control Mode when it is busy but has not yet detected a "match" between the two processors. A match occurs, for example, if Processor A issues a Write Command and Processor B, after sensing, issues a Read Command.

4.2.3 Data Mode - The Channel Adapter is considered to be in the Data Mode upon recognition of a match. When in this mode and transferring data, the Adapter appears busy to both processors until the transfer operation is terminated. It responds with a Busy Status to all commands except the Test I/O Override Command.

Termination of the operation in progress can be accomplished by a Command Out response to a Service In signal, a Selective Reset condition, a Test I/O Override Command, or an Interface Disconnect Sequence. When a Command Out response to a Service In signal (from either processor) is detected in the Adapter, it sends a status byte containing Channel End and Device End information to both processors.

When a Selective Reset condition occurs, Channel End, Device End, and Unit Check will be sent to the processor not causing the Selective Reset. A sense byte containing Equipment Check information will also be sent to that processor.

If a Test I/O Override Command is sent to the Channel Adapter when it is operating in the Data Mode, it replies with a status byte containing Channel End and Device End. It then sends a status byte and a sense byte to the opposite processor. The status byte has Channel End, Device End, and Unit Check Set; the sense byte has the reject bit set.

If the Channel Adapter detects an Interface Disconnect Sequence while operating in the Data Mode, it sends Channel End and Device End Status to both processors. A Unit Check will also be sent if an error condition occurred.

4.2.4 Error Conditions - This paragraph describes the operation of the Channel Adapter under three error conditions. Each error condition is defined and the related operation described in the space below.

Command Parity Errors - The type of error occurs on commands during the Initial Selection Sequence. The Adapter sends a Unit Check and a Bus Out Check to the processor in error. The opposite processor is sent both a Channel End, Device End, and Unit Check Status and a sense byte containing Equipment Check. If the Adapter is not in the Data Mode when the error occurs, it will send the same sense information to the opposite processor but the status byte will contain only Unit Check and Attention Status. A Bus Out Check causes immediate termination in all cases.

Data Parity Errors - This type of error occurs during data transfer. Upon detection of a parity error, the Adapter sends status and sense bytes to both processors. The status byte contains Unit Check, Channel End, and Device End. The sense byte has the Data Check bit set. The originating processor will resend the last byte as soon as it receives the Data Check.

4.2.5 NORMAL COMMAND SEQUENCE FOR DATA TRANSFER

Read Command From Processor A	Initial Selection Sequence
"0" Status to Processor A	
Attention Status to Processor B	Request in Status
Sense Command From Processor B	Initial Selection Sequence
"0" Status to Processor B	
Sense Byte to B (Sense Byte Contains Read Command from A)	Request in Data
Status to Processor B (Channel End, Device End)	Request in Status
Write Command From Processor B	Initial Selection Sequence
"0" Status to Processor B	
Data Request to Processor B (Output Byte from B)	Request in Data
Data Request to Processor A (Input Byte from A)	Request in Data
Data Request to Processor B (Output Byte from B)	Request in Data
Data Request to Processor A (Input Byte to A)	Request in Data
*Data Request to Processor B (Command out Response to Service In (terminate)	Request in Data
Status to Processor A & B (Channel End, Device End)	Request in Status

*Terminate could occur from Processor A or B

4.2.6 Normal Sequence for Control Command Transfers

Control Command from Processor A	Initial Selection Sequence
Status to Processor A (Channel End)	
Attention Status to Processor B	Request in Status
Sense Command from Processor B	Initial Selection Sequence
"0" Status to Processor B	
Sense Byte to B (Sense Byte Contains Control Command from A)	Request in Data
Status to Processor B (Channel End, Device End)	Request in Status
Status to Processor A (Device End)	Request in Status

4.2.7 This paragraph consists of three tablse which define and illustrate the different Command, Status, and Sense Bytes.

Where: P = ODD Parity Bit

D = Command Detail Bit

TABLE 1. Command Repertoire

COMMANDS	P 0 1 2 3 4 5 6 7	PURPOSE OF COMMAND
READ	P D D D D D D 1 0	Conditions the adapter to generate Attention Status to the opposite channel. A sense then a write command would be appropriate response to Attention.
READ BACKWARD	P D D D D 1 1 0 0	Conditions the adapter to generate Attention Status to the opposite channel. A sense then a write command would be the appropriate response to Attention.
WRITE	P D D D D D D 0 1	Conditions the adapter to generate Attention Status to the opposite channel. A sense then a Read Command would be the appropriate response to Attention.
CONTROL	P D D D D D D 1 1	Conditions the adapter to generate Attention Status to the opposite channel. A sense command would be the appropriate response to Attention. At least one of the detail bits must be set.
SENSE	P D D D D 0 1 0 0	Conditions the adapter to input <u>one</u> sense byte to the channel that issued the command.
TEST I/O	P 0 0 0 0 0 0 0 0	Command used to obtain the present status of the channel adapter. Command does not cause generation of new status.
TEST I/O OVERRIDE	P 0 1 0 0 0 0 0 0	Command used to clear existing conditions in the adapter. (Multiplexer Channel only).
NO-OP	P 0 0 0 0 0 0 1 1	Used in test programs to exercise the minimum amount of control unit hardware.

TABLE 2. Status Byte Repertoire

STATUS BYTES	P 0 1 2 3 4 5 6 7	PURPOSE OF STATUS BYTE
ATTENTION	P 1 0 0 0 0 0 0 0	To inform the processor that a command is being held in the adapter register. (Sense command must be issued to obtain command byte).
BUSY	P 0 0 0 1 0 0 0 0	Indicates that a previously initiated command is still in progress or that the adapter was holding pending status.
CHANNEL END	P 0 0 0 0 1 0 0 0	Indicates that part of an operation is complete.
DEVICE END	P 0 0 0 0 0 1 0 0	Indicates completion of an operation.
UNIT CHECK	P 0 0 0 0 0 0 1 0	Indicates an error condition has been detected by the adapter. The sense byte is conditioned to indicate this error.

TABLE 3. Sense Byte Repertoire (Errors)

SENSE BYTES	P 0 1 2 3 4 5 6 7	PURPOSE OF SENSE BYTE
COMMAND REJECT	P 1 0 0 0 0 0 0 0	Indicates that an Invalid Command was detected by the Channel Adapter.
BUS OUT CHECK	P 0 0 1 0 0 0 0 0	A parity error exists on a command.
EQUIPMENT CHECK	P 0 0 0 1 0 0 0 0	Indicates that a Selective Reset or Bus Out Check occurred on the opposite processor.
DATA CHECK	P 0 0 0 0 1 0 0 0	Indicates that a Channel Parity error existed in the last data transfer. Data Check will be presented to both processors.
REJECT	P 0 0 0 0 0 0 1 0	Indicates that a Test I/O Override was generated by the opposite processor or an Interface Disconnect occurred on the opposite processor before a "Match".